Dynamic Optimization of UV Flash Processes

Ritschel, Tobias Kasper Skovborg; Capolei, Andrea; Jørgensen, John Bagterp

Published in:
Proceedings of FOCAPO / CPC 2017

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

DYNAMIC OPTIMIZATION OF UV FLASH PROCESSES

Tobias K. S. Ritschel, Andrea Capolei and John Bagterp Jørgensen*

Department of Applied Mathematics and Computer Science & Center for Energy Resources Engineering, Technical University of Denmark, DK-2800 Kgs Lyngby, Denmark

Abstract

UV flash processes, also referred to as isoenergetic-isochoric flash processes, occur for dynamic simulation and optimization of vapor-liquid equilibrium processes. Dynamic optimization and nonlinear model predictive control of distillation columns, certain two-phase flow problems, as well as oil reservoirs with significant compositional and thermal effects may be conducted as dynamic optimization of UV flash processes. The dynamic optimization problem involving a UV flash problem is formulated as a bilevel optimization problem. This problem is solved using a gradient based single-shooting method. The gradients are computed using the adjoint method and different off-the-shelf optimization software (fmincon, IPOPT, KNITRO, NPSOL) are used for the numerical optimization. Computational results are reported for a flash process involving benzene, toluene and diphenyl. The computational experiments demonstrate that the optimization solver, the compiler, and high-performance linear algebra software are all important for efficient dynamic optimization of UV flash processes.

Keywords

Optimization, Optimal Control, Differential Algebraic Equations, Vapor-Liquid Equilibrium

1 Introduction

Dynamic optimization of vapor-liquid equilibrium (VLE) processes are used in operation and control of distillation columns, certain two-phase flow problems, and oil reservoirs with significant thermal and compositional effects. Therefore, such processes require efficient computational methods for dynamic optimization of UV flash processes. The UV flash problem is also known as the isoenergetic-isochoric flash problem or the UVn flash problem. UVn refers to specification of the internal energy, U, the total volume, V, and the total material amount (moles), n. The second law of thermodynamics, i.e. the entropy of a closed system is maximal, is used to determine the equilibrium composition with U, V, and n specified (Michelsen, 1999). The UV flash problem is different from the more common PT flash problem that occurs in steady-state optimization problems. However, it can be demonstrated that the PT flash problem with additional constraints on the internal energy, U, and the volume, V, is equivalent to the UV flash problem. Algorithmic oriented approaches to dynamic optimization of VLE processes use a nested method in which PT flash problems are solved in the inner loop, and outer loops converge the internal energy, U, and volume, V, to their specified values. Such approaches suffer from computational inefficiency and complicated computations for the gradients. Alternatively, simultaneous methods (Biegler, 2010), multiple- or single-shooting methods (Capolei and Jørgensen, 2012) may be used for dynamic optimization of UV flash processes. In this paper, we present a novel algorithm for dynamic optimization of UV flash processes. The algorithm is based on the single-shooting method and an adjoint method is used for computation of the gradients (Jørgensen, 2007). The numerical integration of the semi-explicit index-1 differential algebraic (DAE) system is the key computational operation in the single-shooting method. We report numerical results for a three-component dynamic UV flash as well as the computational performance for implementations in C and Matlab using different optimization software, different linear algebra software, and different compilers.

*Corresponding author *jbjo@dtu.dk*. Funded by the Danish Advanced Technology Foundation (OPTION; 63-2013-3).

2 Optimal Control Problem

We consider the following optimal control problem (OCP)

$$\min_{[x(t);y(t);z(t)]_{t_0}^{t_f},\{u_k\}_{k\in\mathcal{N}}} \quad \phi = \phi\left([y(t);u(t);d(t)]_{t_0}^{t_f}\right) \tag{1a}$$

subject to

$$x(t_0) = \hat{x}_0, \tag{1b}$$
$$G(x(t), y(t), z(t)) = 0, \quad t \in \mathcal{T}, \tag{1c}$$
$$\dot{x}(t) = F(y(t), u(t), d(t)), \ t \in \mathcal{T}, \tag{1d}$$
$$u(t) = u_k, \ t \in [t_k, \ t_{k+1}[, \quad k \in \mathcal{N}, \tag{1e}$$
$$d(t) = \hat{d}_k, \ t \in [t_k, \ t_{k+1}[, \quad k \in \mathcal{N}, \tag{1f}$$
$$\{u_k\}_{k\in\mathcal{N}} \in \mathcal{U}, \tag{1g}$$

where the objective function is in Lagrange form

$$\phi = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t))dt. \tag{2}$$

$x(t)$ is the state vector, $y(t)$ is a vector of algebraic variables, and $z(t)$ is a vector of adjoint algebraic variables. The estimated initial states, $\hat{x}_0$, and the predicted disturbances, $\{\hat{d}_k\}_{k\in\mathcal{N}}$, are parameters in the optimization problem. $[x(t);\ y(t);\ z(t)]_{t_0}^{t_f}$ is a vector of dependent decision variables, whereas $\{u_k\}_{k\in\mathcal{N}}$ are independent decision variables. The time interval is $\mathcal{T} = [t_0,\ t_f]$ and the control indices are $\mathcal{N} = \{0, 1, \ldots, N-1\}$.

The OCP (1) includes algebraic constraints (1c) and differential equations (1d). The algebraic constraints (1c) are formulated such that they can be used to model equilibrium processes, e.g. VLE processes. Equilibrium processes can be formulated as optimization problems and (1c) can represent the Karush-Kuhn-Tucker (KKT) conditions of these optimization problems. The differential equations (1d) are obtained from conservation principles and the states, $x(t)$, represent the conserved quantities. The right-hand-side in (1d) depends on the algebraic variables, $y(t)$, which are implicit functions of the states through the algebraic constraints (1c), i.e. $y(t) = y(x(t))$. By this statement, we assume that given $x(t)$, it is possible to compute $y(t) = y(x(t))$ and $z(t) = z(x(t))$ by solving $G(x(t), y(t), z(t)) = 0$. This is true for the VLE processes considered in this work. We define the single shooting objective ψ by

$$\psi = \psi(\{u_k\}_{k\in\mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k\in\mathcal{N}}) = \left\{\phi : \text{ (1b)-(1f) } \right\}. \tag{3}$$

Given $\{u_k\}_{k\in\mathcal{N}}$, $\hat{x}_0$, and $\{\hat{d}_k\}_{k\in\mathcal{N}}$, ψ is computed as the objective function, ϕ, obtained by integrating (2) using the solution of (1c)-(1d) with $x(t_0) = \hat{x}_0$, $u(t) = u_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, and $d(t) = \hat{d}_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, i.e. (1b) and (1e)-(1f). Figure 1 illustrates the discretization of the inputs, u, and the numerical computation of the continuous states, x. This is the principle in the single-shooting method and the principle that is used for computation of ψ. With ψ defined by (3), the OCP (1) with the objective function (2) can be expressed as the finite dimensional constrained optimization problem

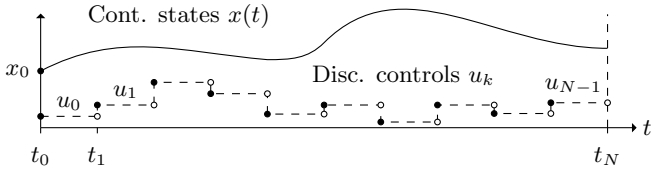


Figure 1. Sketch of the single shooting principle. The controls, $u(t)$, are discretized in time and the continuous states, $x(t)$, are considered functions of the controls. The objective is evaluated by solving the semi-explicit DAEs for a given initial value, x_0, and a given set of controls, $\{u_k\}_{k=0}^{N-1}$.

$$\min_{\{u_k\}_{k\in\mathcal{N}}} \quad \psi = \psi\left(\{u_k\}_{k\in\mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k\in\mathcal{N}}\right), \tag{4a}$$
$$\text{s.t.} \quad \{u_k\}_{k\in\mathcal{N}} \in \mathcal{U}. \tag{4b}$$

The set $\mathcal{U}$ is often a polyhedron such that the constraints (4b) can be expressed by $u_{\min} \leq u \leq u_{\max}$ and $b_l \leq Au \leq b_u$ where $u = [u_0;\ u_1;\ \ldots;\ u_{N-1}]$. Gradient-based optimization algorithms for solution of the nonlinear program (4), and thus the optimal control problem (1), require evaluation of the objective function, ψ, and the gradients, $\{\nabla_{u_k}\psi\}_{k\in\mathcal{N}}$. These computations involve numerical solution of the differential-algebraic equations (DAE), (1c) and (1d), along with computation of the integral (2).

2.1 Equilibrium Constraints

The equilibrium processes that are considered in this work can be described as the solution to a parametric equality constrained optimization problem in the following form

$$\min_{y} \quad f(y) \tag{5a}$$
$$\text{s.t.} \quad g(y) = x, \tag{5b}$$
$$h(y) = 0. \tag{5c}$$

The Langrange function associated with the equilibrium optimization problem (5) is

$$\mathcal{L}(y,\eta,\mu;x) = f(y) - \eta^T(g(y)-x) - \mu^T h(y), \tag{6}$$

where η and μ are Lagrange multipliers associated with (5b) and (5c), respectively. The KKT conditions (first order optimality conditions) for a minimizer $(y = y(x), \eta = \eta(x), \mu = \mu(x))$ are

$$\nabla_y \mathcal{L}(y,\eta,\mu;x) = \nabla f(y) - \nabla g(y)\eta - \nabla h(y)\mu = 0, \tag{7a}$$
$$\nabla_\eta \mathcal{L}(y,\eta,\mu;x) = -(g(y)-x) = 0, \tag{7b}$$
$$\nabla_\mu \mathcal{L}(y,\eta,\mu;x) = -h(y) = 0. \tag{7c}$$

By introducing the vector $z = [\eta;\, \mu]$, we can rewrite the system (7) as the algebraic constraints (1c).

3 Numerical Method

The computation of (3) requires solution of the semi-explicit differential-algebraic initial value problem (1b)-(1d). Subsequently, when $(y(t), u(t), d(t))$ is given, $\psi = \phi$ is computed by quadrature. The system (1b)-(1d) is stiff. Therefore, an implicit method must be used for numerically efficent solution of (1b)-(1d). We exemplify the involved numerical computation using Euler's implicit method. However, the principal numerical methods are also applicable with other implicit solvers such as the ESDIRK methods (Kristensen et al., 2004) and BDF based methods (Barton and Lee, 2002). Furthermore, we describe computation of $\nabla_{u_k}\psi$ for $k \in \mathcal{N}$ by an adjoint method (Jørgensen, 2007; Völcker et al., 2011; Capolei and Jørgensen, 2012). These gradients (sensitivities) may also be computed by a forward method.

3.1 Numerical Integration

Define $w = \begin{bmatrix} x;\, y;\, z \end{bmatrix}$ and define the residual function

$$\begin{aligned} R_{k+1} &= R_{k+1}(w_{k+1}) = R_{k+1}(w_{k+1}; x_k, u_k, d_k) \\ &= R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}; x_k, u_k, d_k) \\ &= \begin{bmatrix} D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, d_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix} \end{aligned} \tag{8}$$

for $k \in \mathcal{N}$ with $D_{k+1} = x_{k+1} - x_k - \Delta t_k F(y_{k+1}, u_k, d_k)$. Given $x_0 = \hat{x}_0$, $\{u_k\}_{k=0}^{N-1}$, and $\{d_k = \hat{d}_k\}_{k=0}^{N-1}$, the implicit Euler discretization of (1b)-(1d) corresponds to solving

$$R_{k+1} = R_{k+1}(w_{k+1}) = 0, \quad k \in \mathcal{N} \tag{9}$$

sequentially for $\{w_{k+1}\}_{k=0}^{N-1}$ by marching forward. Equation (9) is solved by an inexact Newton method, i.e. by solving a sequence of linear systems

$$w_{k+1}^{m+1} = w_{k+1}^m - M_{k+1}^{-1} R_{k+1}(w_{k+1}^m), \tag{10}$$

until some convergence criteria is satisfied. The iteration matrix is

$$M_{k+1} = \frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} I & -\Delta t_k \frac{\partial F}{\partial y} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}, \tag{11}$$

where

$$\frac{\partial G}{\partial x} = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix},\ K = \begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} = \begin{bmatrix} \nabla^2_{yy}\mathcal{L} & -\nabla g & -\nabla h \\ -\nabla g^T & 0 & 0 \\ -\nabla h^T & 0 & 0 \end{bmatrix}.$$

K denotes the KKT matrix of the equilibrium conditions (7). The second derivative of the Lagrangian, $\mathcal{L}$, with respect to y is given by

$$\nabla^2_{yy}\mathcal{L} = \nabla^2 f - \sum_i \eta_i \nabla^2 g_i - \sum_i \mu_i \nabla^2 h_i. \tag{12}$$

3.2 Gradients by the Adjoint Method

We substitute the discrete residual function (8) for the differential-algebraic constraints (1c)-(1d) in the function ψ given by (3) to obtain the following single shooting objective function, in which the zero-order-hold parametrization of the input and disturbances (1e)-(1f) have been applied

$$\psi = \psi(\{u_k\}_{k\in\mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k\in\mathcal{N}}) \tag{13a}$$
$$= \Big\{ \phi = \sum_{k\in\mathcal{N}} \Phi_k(y_{k+1}, u_k, \hat{d}_k) \ : \tag{13b}$$
$$x_0 = \hat{x}_0, \tag{13c}$$
$$R_{k+1}(w_{k+1}, x_k, u_k, \hat{d}_k) = 0,\ k \in \mathcal{N}, \tag{13d}$$
$$[x_{k+1}; y_{k+1}; z_{k+1}] = w_{k+1},\ k \in \mathcal{N} \Big\}. \tag{13e}$$

The Lagrange objective (2) is approximated by the sum (13b) in which Φ_k approximates the integral over $[t_k,\, t_{k+1}[$ using the rectangle rule with y_{k+1} (rather than y_k)

$$\Phi_k = \Phi_k(y_{k+1}, u_k, \hat{d}_k) = \Delta t_k \Phi(y_{k+1}, u_k, \hat{d}_k). \tag{14}$$

The adjoints, $\{\lambda_k\}_{k=1}^N$, are computed by marching backwards in the equations

$$\left(\frac{\partial R_N}{\partial w_N}\right)^T \lambda_N = -\nabla_{w_N}\Phi_{N-1}, \tag{15a}$$
$$\left(\frac{\partial R_k}{\partial w_k}\right)^T \lambda_k = -\left(\frac{\partial R_{k+1}}{\partial w_k}\right)^T \lambda_{k+1} - \nabla_{w_k}\Phi_{k-1}, \tag{15b}$$

for $k \in \{N-1, N-2, \ldots, 1\}$. The Jacobian of the discrete residual $\frac{\partial R_{k+1}}{\partial w_{k+1}}(w_{k+1}, w_k, u_k, \hat{d}_k)$ was defined in (11) and the Jacobian with respect to the states and the algebraic variables in the previous timestep is

$$\frac{\partial R_{k+1}}{\partial w_k}(w_{k+1}, w_k, u_k, \hat{d}_k) = -\begin{bmatrix} \mathrm{I} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{16}$$

for $k = 1, \ldots, N-1$. The gradient of the objective is

$$\nabla_{w_{k+1}} \Phi_k = \begin{bmatrix} 0 \\ \nabla_{y_{k+1}} \Phi_k \\ 0 \end{bmatrix}, \quad k \in \mathcal{N}. \tag{17}$$

The gradients of ψ with respect to the inputs, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$, are computed by

$$\nabla_{u_k} \psi = \nabla_{u_k} \Phi_k + \left(\frac{\partial R_{k+1}}{\partial u_k} \right)^T \lambda_{k+1}, \quad k \in \mathcal{N}. \tag{18}$$

4 UV Flash Example

Neglecting kinetic and potential energy, the energy- and mass balance of a flash unit may be expressed as

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \tag{19a}$$
$$\dot{n_i}(t) = f_{F,i}^v(t) + f_{F,i}^l(t) - v_i(t) - l_i(t), \tag{19b}$$

for the components $i = 1, .., N_C$. U is the internal energy and n_i is the total holdup of component i. H_V and H_L are the enthalpies of the vapor and liquid streams respectively, and H_F^v and H_F^l are the vapor and liquid enthalpies of the feed. $f_{F,i}^v$ and $f_{F,i}^l$ are the vapor and liquid component flow rates of the feed. v_i and l_i are the component flow rates of the vapor and liquid streams. The conservation equations (19) are in the form of the differential equation (1d) where the function F is

$$F(y(t), u(t), d(t)) = \begin{bmatrix} H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t) \\ f_F^v(t) + f_F^l(t) - v(t) - l(t) \end{bmatrix}. \tag{20}$$

Let the state variables, x, the algebraic variables, y, the Lagrange multipliers, z, the manipulated variables, u, and the disturbance variables, d, be defined as:

$$x = [U; n] \in \mathbb{R}^{1+N_C}, \tag{21a}$$
$$y = \left[T; P; n^v; n^l\right] \in \mathbb{R}^{2+2N_C}, \tag{21b}$$
$$z = [\mu; \eta] \in \mathbb{R}^{2+N_C}, \tag{21c}$$
$$u = [Q; F_V; F_L] \in \mathbb{R}^3, \tag{21d}$$
$$d = \left[T_F; P_F; f_F^v; f_F^l\right] \in \mathbb{R}^{2+2N_C}. \tag{21e}$$

The VLE in the flash tank is governed by

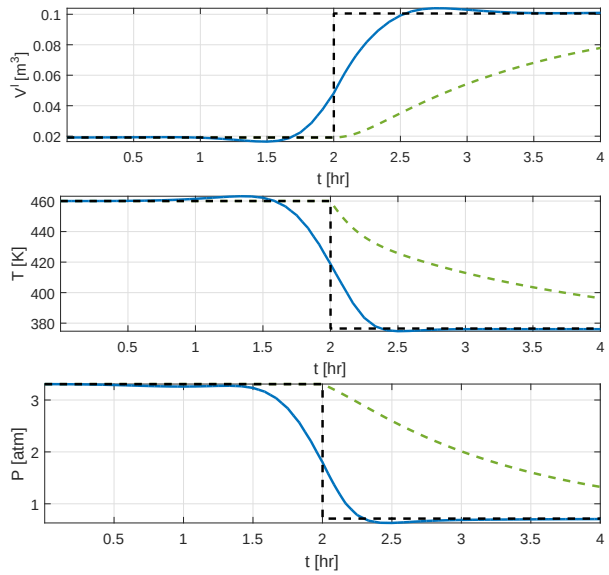
$$\max_{T,P,n^v,n^l} \quad S = S^v(T,P,n^v) + S^l(T,P,n^l) \tag{22a}$$
$$\text{s.t.} \quad U^v(T,P,n^v) + U^l(T,P,n^l) = U, \tag{22b}$$
$$V^v(T,P,n^v) + V^l(T,P,n^l) = V, \tag{22c}$$
$$n_i^v + n_i^l = n_i, \quad i = 1, \ldots, N_C. \tag{22d}$$

The VLE problem (22) is in the form of equation (5) where the functions f, g and h are

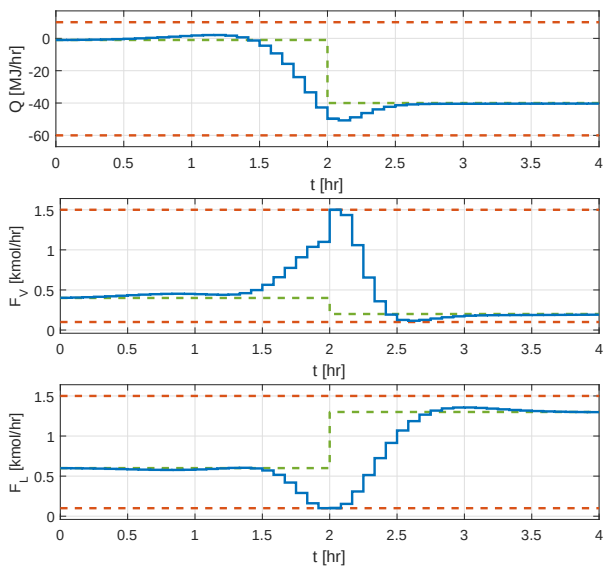
$$f(y) = f(T,P,n^v,n^l) = -\Big(S^v(T,P,n^v) + S^l(T,P,n^l)\Big), \tag{23a}$$
$$g(y) = g(T,P,n^v,n^l) = \begin{bmatrix} U^v(T,P,n^v) + U^l(T,P,n^l) \\ n^v + n^l \end{bmatrix}, \tag{23b}$$
$$h(y) = h(T,P,n^v,n^l) = V^v(T,P,n^v) + V^l(T,P,n^l) - V, \tag{23c}$$

We consider a mixture of benzene, toluene and diphenyl that is separated in a flash tank. Figure 2 shows the controlled variables and the manipulated variables for a least-squares optimal transition between two steady states. The optimal transition is computed by dynamic optimization and compared to an open-loop non-optimized transition. The optimized transition is significantly faster than the transition based on the steady state values of the manipulated variables. Figures 3-5 show the composition variables, selected thermodynamic functions (H, S, G), and the state variables for the optimal transition.

Table 1 shows the computation time for solving the OCP (4) with the presented dynamic optimization algorithm using different compilers, optimization libraries, and linear algebra libraries. fmincon is at least 10 times faster when used with a compiled C implementation for numerical integration compared to a Matlab implementation for numerical integration. When the compiled C code is called from Matlab, it will in all cases be using Intel MKL. Using IPOPT gives a modest speedup of between 6 and 8, because IPOPT uses a limited-memory BFGS update strategy tailored for large-scale systems. KNITRO results in a speedup of between 41 and 48 compared to a pure Matlab implementation, and NPSOL gives a speedup of between 47 and 66. The Intel compilers and Intel MKL generally have a positive effect on the implementations using KNITRO and NPSOL. The compiler icc (Intel) rather than the compiler gcc and Intel MKL rather than Netlib's BLAS/LAPACK have a negative effect on the implementation using IPOPT.



(a) Controlled variables (CV).



(b) Manipulated variables (MV).

Figure 2. Transition between two steady states by dynamic optimization (blue) and use of steady state inputs (green dashed).

5 Conclusion

We presented an adjoint single-shooting algorithm for gradient-based dynamic optimization of flash processes. The algorithm simultaneously solves the equilibrium conditions and the differential conservation equations. A simulation example demonstrates that dynamic optimization enables fast transition between steady states. This is an important feature for nonlinear model predictive control applications. The numerical experiments show that using a simultaneous numerical integration scheme in the adjoint single shooting algorithm yields faster solution than with a nested numerical integration scheme. This is primarily due to fewer evaluations of the thermodynamic properties. Furthermore, the computational time of the adjoint single-shooting algorithm is compared for four optimization solvers (KNITRO, NPSOL, IPOPT and fmincon [Matlab]) and illustrates that using a compiled language together with an appropriate NLP solver library is essential to good computational performance. KNITRO and NPSOL give significant speedup compared to a pure Matlab implementation. IPOPT is designed for large-scale problems and less appropriate for the small dense problem considered in this work. Furthermore, the Intel compilers in combination with the Intel MKL are generally more efficient than using GNU compilers and Netlib's BLAS/LAPACK distribution. Using the suggested algorithm, we can we can solve a dynamic UV flash optimization problem with 3 components in less than 0.2 seconds.

References

Barton, P. I. and Lee, C. K. (2002). Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Transactions on Modeling and Computer Simulation*, pages 256–289.

Biegler, L. T. (2010). *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM.

Capolei, A. and Jørgensen, J. (2012). Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. In *Proceedings of the 2012 American Control Conference*, pages 295–300.

Jørgensen, J. B. (2007). Adjoint sensitivity results for predictive control, state-and parameter-estimation with nonlinear models. In *European Control Conference (ECC), 2007*, pages 3649–3656.

Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., and Jørgensen, S. B. (2004). An ESDIRK method with sensitivity analysis capabilities. *Computers & Chemical Engineering*, 28(12):2695–2707.

Michelsen, M. L. (1999). State function based flash specifications. *Fluid Phase Equilibria*, 158:617–626.

Völcker, C., Jørgensen, J. B., and Stenby, E. H. (2011). Oil reservoir production optimization using optimal control. In *Conference on Decision and Control*, pages 7937–7943.

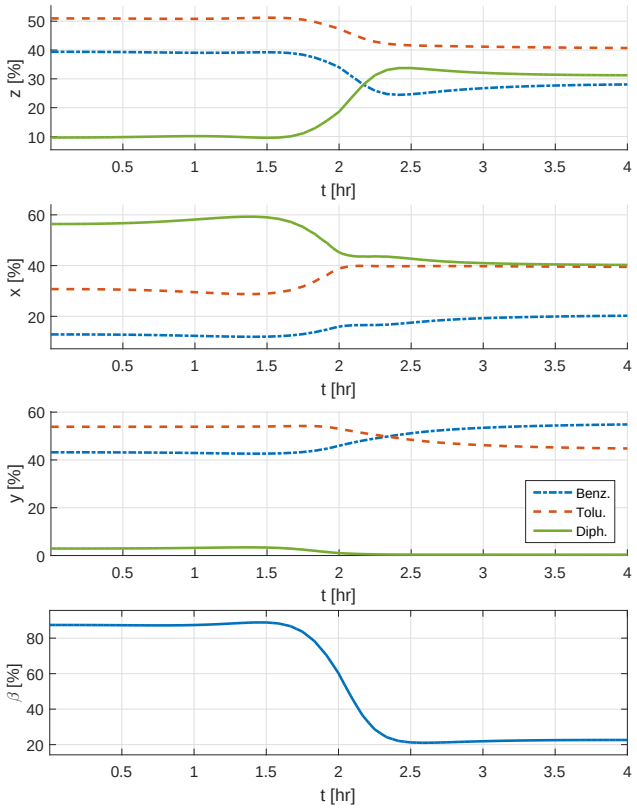


Figure 3. Composition variables of the optimal transition. Overall, z, liquid, x, and vapor, y, mole fractions as well as the vapor fraction, β.

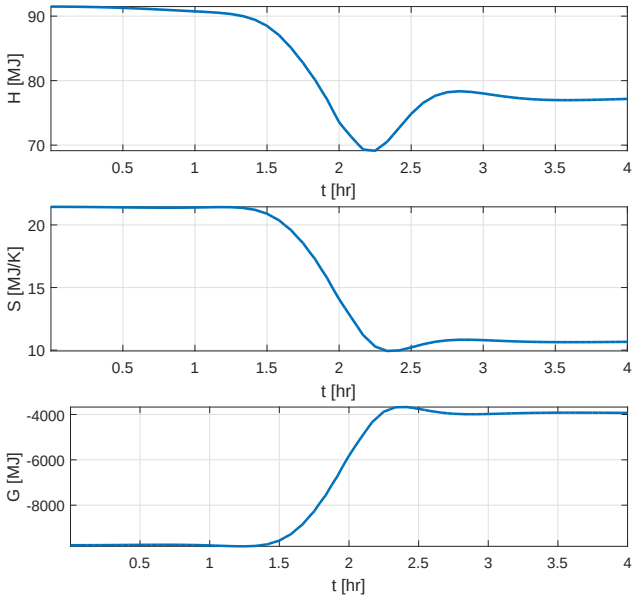


Figure 4. Thermodynamic state functions for the flash tank in the optimal transition. H is the enthalpy, S is the entropy, and G is Gibbs' free energy.

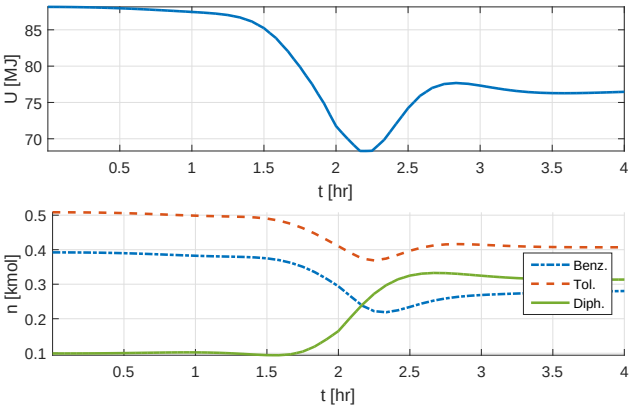


Figure 5. State variables for the optimal transition.

Table 1. Absolute (in seconds) and relative computation time of solving the OCP (4) using simultaneous numerical integration. Average over 10 calls. fmincon (C) uses a C implementation of simultaneous numerical integration and fmincon (M) uses a Matlab implementation. Sp.up is short for speed-up compared to the pure Matlab implementation, i.e. fmincon (M).

	fmincon		IPOPT	KNITRO	NPSOL
	Matlab	C	C	C	C
Iter.	192	192	445	168	158
Func.	195	195	1435	171	159
gcc, gfortran, Netlib BLAS/LAPACK					
Abs.	12.461	1.185	1.663	0.298	0.263
Rel.	1.000	0.095	0.133	0.024	0.021
Sp.up	1.0	10.5	7.5	41.8	47.4
icc, gfortran, Netlib BLAS/LAPACK					
Abs.	12.461	1.081	1.753	0.277	0.246
Rel.	1.000	0.087	0.141	0.022	0.020
Sp.up	1.0	11.5	7.1	45.0	50.7
icc, gfortran, Intel MKL					
Abs.	12.461	1.138	1.876	0.277	0.213
Rel.	1.000	0.091	0.151	0.022	0.017
Sp.up	1.0	10.9	6.6	45.0	58.5
icc, ifort, Intel MKL					
Abs.	12.461	1.149	1.618	0.262	0.189
Rel.	1.000	0.092	0.130	0.021	0.015
Sp.up	1.0	10.8	7.7	47.6	65.9